Rotation, Scaling and Deformation Invariant Scattering for Texture Discrimination

Laurent Sifre
CMAP, Ecole Polytechnique
91128 Palaiseau

Stéphane Mallat
DI, Ecole Normale Superieur
45 rue d'Ulm, 75005 Paris *

Abstract

An affine invariant representation is constructed with a cascade of invariants, which preserves information for classification. A joint translation and rotation invariant representation of image patches is calculated with a scattering transform. It is implemented with a deep convolution network, which computes successive wavelet transforms and modulus non-linearities. Invariants to scaling, shearing and small deformations are calculated with linear operators in the scattering domain. State-of-the-art classification results are obtained over texture databases with uncontrolled viewing conditions.

1. Introduction

Projections of three dimensional surfaces can locally be approximated by affine transforms in the image plane. These affine transformations are multidimensional sources of variability, which may carry little information for classification. Hierarchical cascade of invariants [2, 3] have been studied to build affine invariant image representations. Deep neural networks provide an architecture to compute such invariants, with a succession of linear filters and "pooling" non-linearities, which are learned from data [4, 5].

Learning is not necessary to build affine invariants with a hierarchical cascade, but its mathematical and algorithmic implementation raises difficulties. How can we factorize affine invariants into simpler invariants computed over smaller subgroups ? How to compute stable and informative invariants over any given group ?

This paper shows that stable and informative affine invariant representations can be obtained with a scattering operator defined on the translation, rotation and scaling groups. It is implemented by a deep convolution network with wavelets filters and modulus non-linearities. We study applications to the classification of image textures mapped on three dimensional surfaces.

Section 2 shows that within image patches, translations and rotation invariants must be computed together to retain joint information on spatial positions and orientations. A joint scattering invariant on the roto-translation group requires to build a wavelet transform on this non-commutative group, which involves a different type of convolution described in Section 3. A scaling invariant may however be computed separately with a scale-space averaging across image patches.

Perspective effects also produce non-affine deformations. Thanks to wavelet localizations, scattering transforms computes invariants which are stable to deformations [9]. It results that small shearing and deformations are linearized in the scattering domain. Invariants to these deformations can thus be calculated with linear projectors. Section 5 explains how to optimize such linear projectors for each image class, with a supervised learning. For texture classification, it is implemented with a generative PCA classifier. Section 6 shows that scattering representations give state-of-the-art texture classification results on KTH-TIPS [17], UIUC [18] and UMD [19] databases. All computations can be reproduced by a software available at *www.di.ens.fr/scattering*.

2. Hierarchical Affine Invariants

Section 2.1 analyzes the construction of affine invariants as a a cascade of separable or joint invariants on smaller groups. The hierarchical architecture of affine invariant scattering representations is described in Section 2.2.

2.1. Separable Versus Joint Invariants

The affine group can be written as a product of the translation, rotation, scaling and shearing groups. Affine invariant representations can be computed as a separable product of invariants on each of these smaller subgroups. However, we show that such separable invariant products may lose important information.

The action of an affine operator g of $\mathbb{R}^2$ on an image $x(u)$ yields a warped image $g.x(u) = x(g^{-1}u)$. A repre-

*Work supported by ANR 10-BLAN-0126 and Advanced ERC InvariantClass 320959

sentation $R(x)$ of x is invariant to the action of G if it is not modified by the action of any $g \in G$: $R(g.x) = R(x)$. It is covariant to G if $R(g.x) = g.R(x)$, where g acts on $R(x)$ by shifting its coefficients. A separable invariant on a group product $G = G_1 \rtimes G_2$ combines a first operator R_1, which is invariant to the action of G_1 and covariant to the action G_2, with a second operator R_2 which is invariant to the action of G_2. Indeed for all $g_1.g_2 \in G_1 \rtimes G_2$ and all images $x(u)$:

$$R_2(R_1(g_1.g_2.x)) = R_2(g_2.R_1(x)) = R_2(R_1(x)) .$$

However, such separable invariants do not capture the joint property of the action of G_2 relatively to G_1, and may lose important information. This is why two-dimensional translation invariant representations are not computed by cascading invariants to horizontal and vertical translations. It is also important for rotations and translations. Let us consider for example the two texture patches of Figure 1. A separable product of translation and rotation invariant operators can represent the relative positions of the vertical patterns, and the relative positions of the horizontal patterns, up to global translations. However, it can not represent the positions of horizontal patterns relatively to vertical patterns, because it is not sensitive to a relative shift between these two sets of oriented structures. It loses the relative positions of different orientations, which is needed to be sensitive to curvature, crossings and corners. Such a separable invariant thus can not discriminate the two textures of Figure 1.

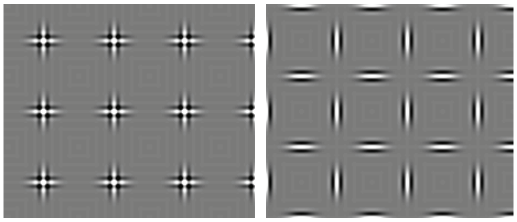

Figure 1: The left and right textures are not discriminated by a separable invariant along rotations and translations, but can be discriminated by a joint roto-translation invariant.

Several authors [6, 7, 8] have proposed to take into account the *joint* structure of roto-translation operators in image processing, particularly to implement diffusion operators. Computing a joint invariant between rotations and translations also means taking into account the joint relative positions and orientations of image structures, so that the textures of Figure 1 can be discriminated. Section 3 introduces a roto-translation scattering operator, which is computed by cascading wavelet transforms on the roto-translation group.

Calculating joint invariants on large non-commutative groups may however become very complex. Keeping a separable product structure is thus desirable as long as it does not lose too much information. This is the case for scaling. Indeed, local image structures are typically spread across scales, with a power law decay. This is the case for contours, singularities and most natural textures. As a result of this strong correlation across scales, one can use a separable invariant along scales, with little loss of discriminative information.

2.2. Hierarchical Architecture

We now explain how to build an affine invariant representation, with a hierarchical architecture. We separate variabilities of potentially large amplitudes such as translations, rotations and scaling, from smaller amplitude variabilities, but which may belong to much higher dimensional groups such as shearing and general diffeomorphisms. These small amplitude deformations are linearized to remove them with linear projectors.

Image variabilities typically differ over domains of different sizes. Most image representations build localized invariants over small image patches, for example with SIFT descriptors [15]. These invariant coefficients are then aggregated into more invariant global image descriptors, for example with bag of words [10] or multiple layers of deep neural network [4, 5]. We follow a similar strategy by first computing invariants over image patches and then aggregating them at the global image scale. This is illustrated by the computational architecture of Figure 2.

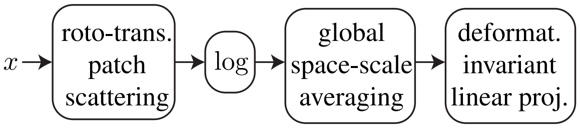
x
roto-trans. patch scattering
log
global space-scale averaging
deformat. invariant linear proj.

Figure 2: An affine invariant scattering is computed by applying a roto-translation scattering on image patches, a logarithmic non-linearity and a global space-scale averaging. Invariants to small shearing and deformations are computed with linear projectors optimized by a supervised classifier.

Within image patches, as previously explained, one must keep the joint information between positions and orientations. This is done by calculating a scattering invariant on the joint roto-translation group. Scaling invariance is then implemented with a global scale-space averaging between patches, described in Section 4. A logarithmic non-linearity is first applied to invariant scattering coefficients to linearize their power law behavior across scales. This is similar to the normalization strategies used by bag of words [10] and deep neural networks [5].

Because of three dimensional surface curvature in the visual scene, the image patches are also deformed. A scattering transform was proved to be stable to deformations [9]. Indeed, it is computed with a cascade of wavelet transforms which are stable to deformations, because wavelets are both regular and localized. A small image deformation thus produces a small modification of it scattering representation. The stability of scattering transforms to deformations guarantees that small shearing and deformations can be approximated by a linear operator in the space of scattering coefficients. Invariants to small deformations can thus be computed with linear projectors. Enforcing invariance to all deformations would remove too much information because deformations involve too many degrees of freedom. To reduce this information loss, we only compute a subset of deformation invariants, which is adapted to each signal class. Since invariants are implemented by a linear projector, their optimization involves the optimization of a linear operator. This can be done by an Support Vector Machine or other supervised learning algorithms, which perform the classification from an optimized linear combination of the data. Section 5 provides an implementation using a generative Principal Component Analysis (PCA) classifier.

3. Roto-Translation Patch Scattering

This section introduces roto-translation scattering operators which are stable to deformations.

3.1. Invariant-Covariant Wavelet Cascade

A scattering operator [9] computes an invariant image representation relatively to the action of a group, by applying a cascade of invariant and covariant operators calculated with wavelet convolutions and modulus operators. Convolutions on a group appear naturally because they define all linear operators which are covariant to the action of a group. We concentrate on the roto-translation group.

An element $g = (v, \theta)$ of the roto-translation group $G = \mathbb{R}^2 \rtimes SO(2)$ acting on $u \in \mathbb{R}^2$ combines a translation by v and a rotation $r_\theta \in SO(2)$:

$$gu = v + r_\theta u \ . \tag{1}$$

The product of two roto-translations $g = (v, \theta)$ and $h = (v', \theta')$ is

$$gh = (v + r_\theta v', \ \theta + \theta') \ , \tag{2}$$

so the inverse of g is $g^{-1}u = r_{-\theta}(u - v)$. The action of g on an image $x(u)$ translates x by v and rotates it by θ: $g.x(u) = x(g^{-1}u)$.

A scattering representation computes successive layers $U_m x$ of signal coefficients, which are covariant to the action of G. It means that $U_m(g.x) = g.U_m x$, where $g.U_m x$ performs a "translation and rotation" of the coefficients of $U_m x$. Local scattering invariants of order m are computed by averaging $h.U_m x$ for all h in the neighborhood of any $g \in G$:

$$S_m x(g) = \sum_{h \in G} h.U_m x \ \Phi_J(h^{-1}g) \ . \tag{3}$$

This is a convolution on the group G. The support of the averaging filter Φ_J defines the invariance domain. To compute local roto-translation invariants on image patches of size 2^J, we choose $\Phi_J(u', \theta') = (2\pi)^{-1}\phi_J(u')$, where the spatial support of ϕ_J is proportional to 2^J. For $g = (u, \theta)$, the convolution (3) averages $h.U_m x$ over all rotation angles, in a spatial neighborhood of u of size proportional to 2^J.

The average $S_m x$ carries the low frequencies of $U_m x$ relatively to "shifts" along the roto-translation group G, and thus loses all high frequencies. High frequencies are captured by roto-translation convolutions with wavelets. Sections 3.2 and 3.3 introduce a wavelet modulus operator $\widetilde{W}_m$, which transforms $U_m x$ into the average $S_m x$ and a new layer $U_{m+1} x$ of wavelet amplitude coefficients:

$$\widetilde{W}_m U_m x = (S_m x, \ U_{m+1} x) \ . \tag{4}$$

The new layer $U_{m+1} x$ is computed with wavelet roto-translation convolutions and thus remains covariant to roto-translations. Iterating on this wavelet modulus transform outputs multiple layers of scattering invariant coefficients. For $m = 0$ we initialize $U_0 x = x$. Figure 3 illustrates the calculation of a second order scattering vector

$$Sx = (S_0 x, S_1 x, S_2 x) \, , \tag{5}$$

by successively applying $\widetilde{W}_m$ for $1 \le m \le 3$. Next sections define stable and contractive operators $\widetilde{W}_m$, so that the resulting scattering representation Sx is also contractive and stable to deformations.

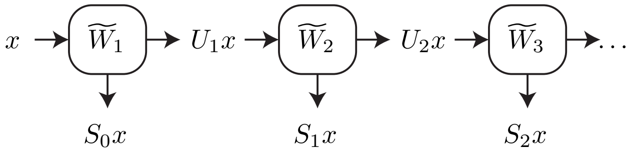

Figure 3: A scattering representation is calculated with a cascade of wavelet-modulus operators $\widetilde{W}_m$. Each $\widetilde{W}_m$ outputs invariant scattering coefficients $S_m x$ and a next layer of covariant wavelet modulus coefficients $U_{m+1} x$, which is further transformed.

3.2. First Layer With Spatial Wavelets

This section defines the first wavelet modulus operator $\widetilde{W}_1$ which computes $S_0 x$ and $U_1 x$ from the input image x.

Locally invariant translation and rotation coefficients are first computed by averaging the image x with a rotation invariant low pass filter $\phi_J(u) = 2^{-2J}\phi(2^{-J}u)$:

$$S_0 x(u) = x \star \phi_J(u) = \sum_v x(v)\phi_J(u - v) \ . \tag{6}$$

In this paper, ϕ is a rotationally invariant gaussian. The averaged image $S_0 x$ is nearly invariant to rotations and translations up to 2^J pixels, but it has lost the high frequencies of x. These high frequencies are recovered by convolution with high pass wavelet filters. To obtain rotation covariant coefficients, we rotate a wavelet ψ by several angles θ and dilate it by 2^j:

$$\psi_{\theta,j}(u) = 2^{-2j}\psi(2^{-j}r_{-\theta}u) \ . \tag{7}$$

The resulting wavelet transform W_1 computes

$$W_1x = (x \star \phi_J(u),\ x \star \psi_{\theta,j}(u))_{u,\theta,j} \ . \tag{8}$$

Wavelets are designed so that W_1 is contractive and potentially unitary [1, 9]. We use complex wavelets whose real and imaginary parts have a quadrature phase. The complex phase of $x \star \psi_{j,\theta}$ then varies linearly with small translations of x. Removing this phase with a modulus operator yields a regular envelop which is more insensitive to translations:

$$U_1x(p_1) = |x \star \psi_{\theta_1,j_1}(u)| \ \text{ with } \ p_1 = (u, \theta_1, j_1) \ . \tag{9}$$

The vector of coefficients U_1x is computed with spatial convolutions and is thus covariant to a translation of x. It is also covariant to rotations $r_{\theta'}x(u) = x(r_{-\theta'}u)$. With a change of variable we indeed verify that

$$U_1(r_\theta x)(u, j_1, \theta_1) = U_1x(r_{-\theta}u, j_1, \theta_1 - \theta),$$

which defines the action of r_θ on U_1x. The resulting wavelet-modulus operator is

$$\widetilde{W}_1x = \Big(x \star \phi_J,\ \{|x \star \psi_{\theta,j}|\}_{\theta,j}\Big) = (S_0x, U_1x). \tag{10}$$

The non-linear operator $\widetilde{W}_1$ is contractive because the wavelet transform W_1 is contractive and a modulus is also contractive. The norm of $\widetilde{W}_1x$ is equal to the norm of x if W_1 is unitary. One can prove that $\widetilde{W}_1$ is stable to deformations because wavelets are regular localized functions [9]. Numerical applications in this paper are calculated with complex Morlet wavelets displayed in Figure 4. They are equal to Gabor functions whose mean is set to zero by subtracting a Gaussian.

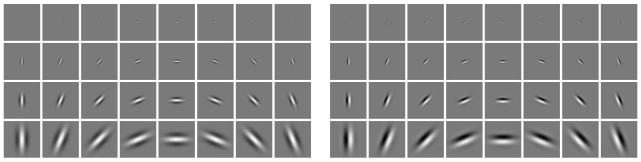

Figure 4: Quadrature phase complex Morlet wavelets $\psi_{j,\theta}$, dilated (along rows) and rotated (along columns). Their real and imaginary parts are shown on the left and on the right, respectively.

3.3. Deeper Layer With Roto-Translation Wavelets

The second wavelet modulus operator $\widetilde{W}_2$ computes the average S_1x of U_1x on the roto-translation group, together with the next layer of wavelet modulus coefficients U_2x. This is performed with convolutions on the roto-translation group G. The convolution of two functions $Y(g)$ and $Z(g)$ defined on G is:

$$Y \circledast Z(g) = \sum_{h \in G} Y(h)Z(h^{-1}g) \,. \tag{11}$$

The invariant part of U_1 is computed with an averaging over the spatial and angle variables. It is implemented for each j_1 fixed, with a roto-translation convolution of $Y(h) = U_1x(h, j_1)$ along the $h = (u', \theta')$ variable, with an averaging kernel $\Phi_J(h)$. For $p_1 = (g_1, j_1)$ and $g_1 = (u, \theta_1)$, this is written

$$S_1x(p_1) = U_1x(.,\, j_1) \circledast \Phi_J(g_1) \ . \tag{12}$$

We choose $\Phi_J(u', \theta') = (2\pi)^{-1}\phi_J(u')$ to perform an averaging over all angles θ and over a spatial domain proportional to 2^J.

The high frequencies lost by this averaging are recovered through roto-translation convolutions with separable wavelets. Roto-translation wavelets are computed with three separable products. Complex quadrature phase spatial wavelets $\psi_{\theta_2,j_2}(u)$ or averaging filters $\phi_J(u)$ are multiplied by complex 2π periodic wavelets $\overline{\psi}_k(\theta)$ or by $\overline{\phi}(\theta) = (2\pi)^{-1}$:

$$\begin{aligned} \Psi_{\theta_2,j_2,k_2}(u,\theta) &= \psi_{\theta_2,j_2}(u)\,\overline{\psi}_{k_2}(\theta) && (13)\\ \Psi_{0,J,k_2}(u,\theta) &= \phi_J(u)\,\overline{\psi}_{k_2}(\theta) && (14)\\ \Psi_{\theta_2,j_2,0}(u,\theta) &= \psi_{\theta_2,j_2}(u)\,\overline{\phi}(\theta) \ . && (15) \end{aligned}$$

A roto-translation scattering iterates on roto-translation wavelet-modulus operator, defined for any $m \geq 2$ and any function $Y(g)$ by

$$\widetilde{W}_mY = \Big(Y \circledast \Phi_J(g),\ |Y \circledast \Psi_{\theta_2,j_2,k_2}(g)|\Big)_{g,\theta_2,j_2,k_2} \ . \tag{16}$$

This wavelet modulus operator is contractive and preserves the norm for appropriate wavelets. It is also stable to deformations because roto-translation wavelets are localized. For $m = 2$, we apply $\widetilde{W}_2$ to $Y(g) = U_1x(g, j_1)$, for j_1 fixed. It computes $\widetilde{W}_2U_1x = (S_1x, U_2x)$, where S_1x is defined in (12) and

$$U_2x(p_2) = |U_1x(., j_1) \circledast \Psi_{\theta_2,j_2,k_2}(g_1)| \tag{17}$$

with $g_1 = (u, \theta_1)$, $p_2 = (g_1, \overline{p}_2)$, and $\overline{p}_2 = (j_1, \theta_2 - \theta_1, j_2, k_2)$. Since $U_2x(p_2)$ is computed with a roto-translation convolution, it remains covariant to the action of the roto-translation group.

Fast computations of roto-translation convolutions with separable wavelet filters $\Psi_{\theta_2,j_2,k_2}(u,\theta) = \psi_{\theta_2,j_2}(u)\overline{\psi}_{k_2}(\theta)$ are performed by factorizing

$$Y \circledast \Psi_{\theta_2,j_2,k_2}(u,\theta) \\ = \sum_{\theta'} \left(\sum_{u'} Y(u',\theta')\, \psi_{\theta_2,j_2}(r_{-\theta'}(u-u')) \right) \overline{\psi}_{k_2}(\theta-\theta') \, .$$

It is thus computed with a two-dimensional convolution of $Y(u,\theta')$ with $\psi_{\theta_2,j_2}(r_{-\theta}u)$ along $u = (u_1,u_2)$, followed by a convolution of the output and a one-dimensional circular convolution of the result with $\overline{\psi}_{k_2}$ along θ. Figure 5 illustrates this convolution which rotates the spatial support $\psi_{\theta_2,j_2}(u)$ by θ while multiplying its amplitude by $\overline{\psi}_{k_2}(\theta)$.

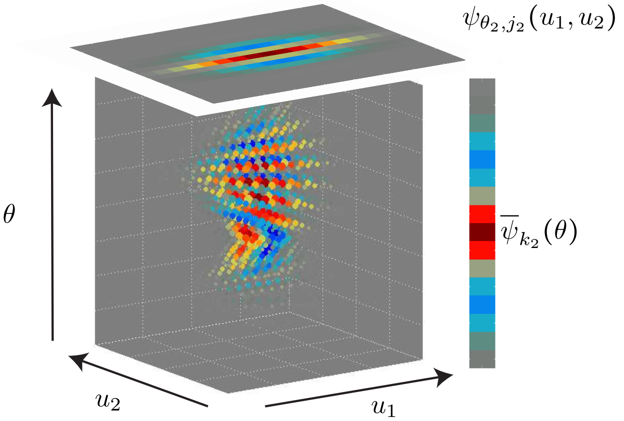
$\psi_{\theta_2,j_2}(u_1,u_2)$
θ
$\overline{\psi}_{k_2}(\theta)$
u_2
u_1

Figure 5: A three dimensional roto-translation convolution with a wavelet $\Psi_{\theta_2,j_2,k_2}(u_1,u_2,\theta)$ can be factorized into a two dimensional convolution with $\psi_{\theta_2,j_2}(u_1,u_2)$ rotated by θ and a one dimensional convolution with $\overline{\psi}_{k_2}(\theta)$.

Applying $\widetilde{W_3} = \widetilde{W_2}$ to U_2x computes second order scattering coefficients as a convolution of $Y(g) = U_2x(g,\overline{p}_2)$ with $\Phi_J(g)$, for $\overline{p}_2$ fixed:

$$S_2x(p_2) = U_2(.,\overline{p}_2)x \circledast \Phi_J(g) \, . \tag{18}$$

It also computes the next layer of coefficients U_3x with a roto-translation convolution of $U_2x(g,\overline{p}_2)$ with the wavelets (13,14,15). In practice, we stop at the second order because the coefficients of U_3x carry a small amount of energy, and have little impact on classification. One can indeed verify that the energy of U_mx decreases exponentially to zero as m increases.

The output roto-translation of a second order scattering representation is a vector of coefficients:

$$Sx = \Big(S_0x(u)\, ,\, S_1x(p_1)\, ,\, S_2x(p_2)\Big)\, , \tag{19}$$

with $p_1 = (u,\theta_1,j_1)$ and $p_2 = (u,\theta_1,j_1,\theta_2,j_2,k_2)$. The spatial variable u is sampled at intervals 2^J which corresponds to the patch size. If x is an image of N^2 pixels, there are thus $2^{-2J}N^2$ coefficients in S_0x and $2^{-2J}N^2J$ coefficients in S_1x. Second order coefficients have a negligible amplitude if $j_2 \leq j_1$. If the wavelet are rotated along K angles θ then one can verify that S_2x has approximatively $2^{-2J}N^2J(J-1)K\log_2 K/2$ coefficients. The total roto-translatation patch scattering Sx is of dimension $341N^2/1024$ for $J=5$ and $K=8$. The overall complexity to compute this roto-translation scattering representation is $O(K^2N^2\log N)$.

4. Scaling Invariance of Log Scattering

Roto-translation scattering is computed over image patches of size 2^J. Above this size, perspective effects produce important scaling variations for different patches. A joint scale-rotation-translation invariant must therefore be applied to the scattering representation of each patch vector. This is done with an averaging along the scale and translation variables, with a filter which is rotationally symmetric. One could recover the high frequencies lost by this averaging and compute a new layer of invariant through convolutions on the joint scale-rotation-translation group. However, adding this supplementary information does not improve texture classification, so this last invariant is limited to a global scale-space averaging.

The roto-translation scattering representations of all patches at a scale 2^J is given by

$$Sx = \Big(x \star \phi_J(u)\, ,\, U_1x \circledast \Phi_J(p_1)\, ,\, U_2x \circledast \Phi_J(p_2)\Big)\, ,$$

with $p_1 = (u,\theta_1,j_1)$ and $p_2 = (u,\theta_1,j_1,\theta_2,j_2,k_2)$. This scattering vector Sx is not covariant to scaling. If $x_i(u) = x(2^iu)$ then

$$Sx_i = \Big(x \star \phi_{J+i}(2^iu) \quad , \quad U_1x \circledast \Phi_{J+i}(2^i.p_1) \\ U_2x \circledast \Phi_{J+i}(2^i.p_2)\Big)\, .$$

with $2^i.p_1 = (2^iu,\theta_1,j_1+i)$ and $2^i.p_2 = (2^iu,\theta_1,j_1+i,\theta_2,j_2+i,k_2)$. A covariant representation to scaling stores the minimal subset of coefficients needed to recover all Sx_i. It thus require to compute the scattering coefficients for all scales j_1+i and j_2+i for all averaging kernels ϕ_{J+i} or Φ_{J+i}, similarly to spatial pyramid [16].

One can show that scattering coefficient amplitudes have a power law decay as a function of the scales 2^{j_1} and 2^{j_2}. To estimate an accurate average from a uniform sampling of the variables j_1 and j_2, it is necessary to bound uniformly the variations of scattering coefficient as a function of j_1 and j_2. This is done by applying a logarithm to each coefficient of Sx, which nearly linearizes the dependency upon j_1 and j_2. This logarithm plays a role which is similar to renormalizations used in bag of words [10] and deep convolution networks [5].

A joint scaling, rotation and translation invariant is computed with a scale-space averaging of $\log Sx_i$ along the scale and spatial indices (i, u):

$$\overline{S}x = \sum_{i,u} \log(Sx_i(u,.))\phi_I(i) \ . \qquad (20)$$

The precision of this averaging is improved by sampling i at half integers. It require to compute twice more scattering coefficients at scales $2^{j_1/2}$ and $2^{j_2/2}$. If 2^I is the length of the averaging kernel $\phi_I(i)$ then 2^{J+2^I} must be smaller than the image size. In texture applications, these averages can only be computed on a small range of scales $2^I = 2$. One could recover the information lost by the scale-space averaging (20) through convolutions with wavelets defined on the joint scale-rotation-translation group, and define a new scattering cascade. This is needed to characterize very large scale texture structures, which is not done in this paper. The invariant image representation $\overline{S}x$ is of dimension 536 if computed over image patches of size $2^J = 2^5 = 32$ with $K = 8$ wavelet orientations. This relatively small feature vector does not depend upon the image size, which is usually larger than 10^5 pixels.

5. Deformation Invariant Projectors

Shearing and image deformations are typically of smaller amplitudes than translations, rotations and scaling. A scattering transform is stable and hence linearizes small deformations. A set of small image deformations thus produces scattering coefficients which belong to an affine space. Linear projectors which are orthogonal to this affine space are invariant to these small deformations. These invariants can be adapted to each signal class by optimizing a linear kernel at the supervised classification stage. This may be done by an SVM but we shall rather use a generative PCA classifier as in [1]. Such classifiers can indeed perform better when the training set is small.

Each signal class is represented by a random vector x_c for $1 \leq c \leq C$, whose realizations are images in the class c. The scattering transform $\overline{S}x_c$ is a random vector. It's expected value is written $\mathbb{E}(\overline{S}x_c)$. A PCA diagonalizes the covariance matrix of $\overline{S}x_c$. Let $\mathbf{V}_c$ be the linear space generated by the D eigenvectors of the covariance matrix of largest eigenvalues. Approximating $\overline{S}x_c - \mathbb{E}(\overline{S}x_c)$ by its projection in $\mathbf{V}_c$ gives a minimum mean-square error, among all projections in linear spaces of dimension D. The space $\mathbf{V}_c$ includes the variability directions produced by deformations of textures in the class. Let $\mathbf{V}_c^\perp$ be its orthogonal complement. The orthogonal projection $P_{\mathbf{V}_c^\perp}$ is an invariant operator which filters out these main intra-class variability. If x is in the class c then $\|P_{\mathbf{V}_c^\perp}(\overline{S}x - \mathbb{E}\overline{S}x_c)\|$ is typically small because most of the energy of $\overline{S}x - \mathbb{E}(\overline{S}x_c)$ is in $\mathbf{V}_c$.

As in [1], we use a simple quadratic classifier which associates to each signal x the class index $\hat{c}$ which minimizes the projected distance to the class centroid:

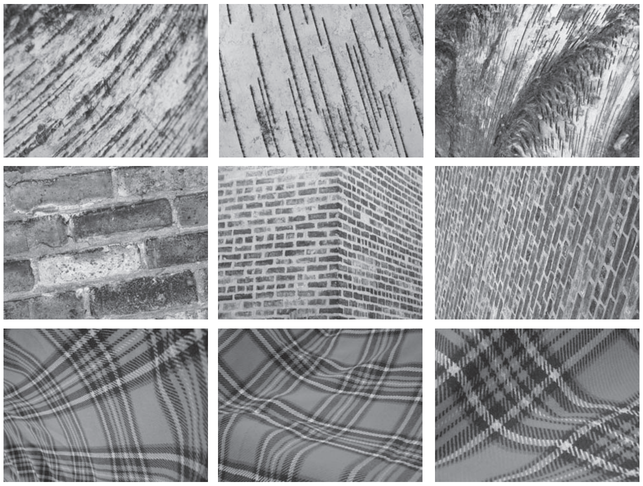

Figure 6: Each row shows images from the same texture class in the UIUC database [10], with important rotation, scaling and deformation variability.

$$\hat{c}(x) = \arg\min_{1 \leq c \leq C} \|P_{\mathbf{V}_c^\perp}(\overline{S}x - \mathbb{E}\overline{S}x_c)\|^2 \ . \qquad (21)$$

It finds the class centroid $\mathbb{E}(\overline{S}x_c)$ which is the closest to $\overline{S}x$, after eliminating the first D principal variability directions.

6. Texture Classification Experiments

This section gives scattering classification results on KTH-TIPS [17], UIUC [10, 18] and UMD [19] texture datasets, and comparison with state of the art algorithms. We first review state of the art approaches based on different types of invariants.

Most state of the art algorithms use separable invariants to define a translation and rotation invariant algorithms, and thus lose joint information on positions and orientations. This is the case of [10] where rotation invariance is obtained through histograms along concentric circles, as well as Log Gaussian Cox processes (COX) [11] and Basic Image Features (BIF) [12] which use rotation invariant patch descriptors calculated from small filter responses. Sorted Random Projection (SRP) [14] replaces histogram with a similar sorting algorithm and adds fine scale joint information between orientations and spatial positions by calculating radial and angular differences before sorting. Wavelet Multifractal Spectrum (WMFS) [13] computes wavelet descriptors which are averaged in space and rotations, and are similar to first order scattering coefficients S_1x.

We compare the best published results [10, 11, 12, 13, 14] and scattering invariants on KTH-TIPS (table 1), UIUC (table 2) and UMD (table 3) texture databases. For each database, Tables 1,2,3 give the mean classification rate and standard deviation over 200 random splits between training and testing for different training sizes. Classification rates are computed with scattering representations implemented with progressively more invariants, and with the PCA classifier of Section 5. As the training sets are small for each class c, the dimension D of the high variability space $\mathbf{V}_c$ is set to the training size. The space $\mathbf{V}_c$ is thus generated by the D scattering vectors of the training set. For larger training databases, it must be adjusted with a cross validation as in [1].

Classification rates in Tables 1,2,3 are given for different scattering representations. The rows "trans. scatt" correspond to a translation invariant scattering as in [1]. It is computed on image patches of size 2^J, with a final spatial averaging of all the patch scattering vector. The rows "roto-trans. scatt" replace the translation invariant scattering by the roto-translation scattering of Section 3. The rows "+ log" show that the error is reduced by adding a logarithmic non-linearity before the spatial averaging. The rows "+ scale avg" also reduce the error by computing a separable invariant along scales, with the averaging described in Section 4. The rows "+ multiscale train" are obtained by augmenting the training set of the PCA. A "scaling" by 2^i of each scattering vector Sx is obtained by shifting the scale indices (j_1, j_2) of Sx by i: $(j_1 + i, j_2 + i)$. This "scaling" is done on the scattering vectors of all training images for multiple values of i. The averaging along the scale variable is not implemented on training samples. It is left to the PCA to choose which projector is best adapted to optimize the classification.

Train size	5	20	40
COX [11]	80.2 ± 2.2	92.4 ± 1.1	95.7 ± 0.5
BIF [12]	-	-	98.5
SRP [14]	-	-	99.3
trans scatt	69.1 ± 3.5	94.8 ± 1.3	98.0 ± 0.8
roto-trans scatt	69.5 ± 3.6	94.9 ± 1.4	98.3 ± 0.9
+ log	76.2 ± 3.3	96.0 ± 1.1	98.8 ± 0.7
+ scale avg	77.8 ± 3.6	97.4 ± 1.0	99.2 ± 0.6
+ multiscal train	**84.3** ± 3.1	**98.3** ± 0.9	**99.4** ± 0.4

Table 1: Classification rates with standard deviations on KTH-TIPS [17] database. Columns correspond to different training sizes per class. The first few rows give the best published results. The last five rows give results obtained with progressively refined scattering invariants. Best results are bolded.

KTH-TIPS contains 10 classes of 81 samples with controlled scaling, shear and illumination variations but no rotation. The roto-translation scattering does not degrade results but each scale processing step provides significant improvements. UIUC (Figure 6) and UMD both contain 25 classes of 40 samples with uncontrolled full affine and illumination variation as well as large elastic deformations. For both these databases, the roto-translation scattering provides a considerable improvement (from 50% to 77% for UIUC with 5 training) and scale processing steps also improve results. The overall approach achieves and often exceeds state-of-the-art results on all these databases. For these three databases, we have used the same filters and options except for the patch size 2^J, which is proportional to the image size. It results that $J = 4$ for KTH-TIPS images which are 200×200, $J = 5$ for UIUC images which are 640×480 and $J = 6$ for UMD images which are 1280×960.

Training size	5	10	20
Lazebnik [10]	-	92.6	96.0
WMFS [13]	**93.4**	97.0	98.6
BIF [12]	-	-	98.8 ± 0.5
trans scatt	50.0 ± 2.1	65.2 ± 1.9	79.8 ± 1.8
roto-trans scatt	77.1 ± 2.7	90.2 ± 1.4	96.7 ± 0.8
+ log	84.3 ± 2.1	94.5 ± 1.1	98.2 ± 0.6
+ scale avg	86.6 ± 2.0	95.4 ± 1.0	98.6 ± 0.6
+ multiscal train	93.3 ± 1.4	**97.8** ± 0.6	**99.4** ± 0.4

Table 2: Classification rates on UIUCTex [10, 18] database.

Training size	5	10	20
WMFS [13]	93.4	97.0	98.7
SRP [14]	-	-	99.3
trans scatt	80.2 ± 1.9	91.8 ± 1.4	97.4 ± 0.9
roto-trans scatt	87.5 ± 2.2	96.5 ± 1.1	99.2 ± 0.5
+ log	91.9 ± 1.7	97.6 ± 0.8	99.3 ± 0.4
+ scale avg	91.6 ± 1.6	97.7 ± 0.9	99.6 ± 0.4
+ multiscal train	**96.6** ± 1.0	**98.9** ± 0.6	**99.7** ± 0.3

Table 3: Classification rates on UMD [19] database.

7. Conclusion

This paper introduces a general scattering architecture which computes invariants to translations, rotations, scaling and deformations, while keeping enough discriminative information. It can be interpreted as a deep convolution network, where convolutions are performed along spatial, rotation and scaling variables. As opposed to standard convolution networks, the filters are not learned but are scaled and rotated wavelets.

State-of-the-art texture discrimination results are obtained on all tested texture databases, including the most difficult ones such as UIUC and UMD, which include important deformations. This paper concentrates on texture applications, but the invariance properties of this scattering image patch representation can also replace SIFT type features for more complex classification problems.

References

[1] J. Bruna, S. Mallat, “Invariant Scattering Convolution Networks”, IEEE Trans. on PAMI, to appear.

[2] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, “Robust object recognition with cortex-like mechanisms,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 29:411-426, 2007.

[3] T. Poggio, J. Mutch, F. Anselmi, L. Rosasco, J.Z. Leibo, and A. Tacchetti, “The computational magic of the ventral stream: sketch of a theory”, MIT-CSAIL-TR-2012-035, December 2012.

[4] G. E. Hinton and R. R. and Salakhutdinov, “Reducing the dimensionality of data with neural networks”, Science, Vol. 313. no. 5786, pp. 504 - 507, 28 July 2006.

[5] Y. LeCun, K. Kavukvuoglu and C. Farabet: “Convolutional Networks and Applications in Vision”, Proc. of ISCAS 2010.

[6] G. Citti, A. Sarti, “A Cortical Based Model of Perceptual Completion in the Roto-Translation Space”, Journal of Mathematical Imaging and Vision archive, Vol. 24, no. 3, p. 307=326, May 2006.

[7] U. Boscain, J. Duplaix, J.P. Gauthier, F. Rossi “Anthropomorphic Image Reconstruction via Hypoelliptic Diffusion”, SIAM Journal on Control and Optimization, to appear.

[8] R. Duits, B. Burgeth, “Scale Spaces on Lie Groups”, in Scale Space and Variational Methods in Computer Vision, Springer Lecture Notes in Computer Science, Vol. 4485, 2007, pp 300-312.

[9] S. Mallat “Group Invariant Scattering”, Communications in Pure and Applied Mathematics, vol. 65, no. 10. pp. 1331-1398, October 2012.

[10] S. Lazebnik, C. Schmid and J. Ponce, “A sparse texture representation using local affine regions”, IEEE Trans. on PAMI, vol. 27, no. 8, pp. 1265-1278, August 2005.

[11] Huu-Giao Nguyen, R, Fablet, and J.-M. Boucher, “Visual textures as realizations of multivariate log-Gaussian Cox processes”, Proc. of CVPR, June 2011

[12] M. Crosier and L.D. Griffin, “Texture classification with a dictionary of basic image features”, Proc. of CVPR, June 2008

[13] Y. Xu, X. Yang, H. Ling and H. Ji, “A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid”, Proc. of CVPR, June 2010

[14] L. Liu, P. Fieguth, G. Kuang, H. Zha, “Sorted Random Projections for Robust Texture Classification”, Proc. of ICCV, 2011

[15] D. Lowe. “Distinctive image features from scale-invariant keypoints”, IJCV, 60(4):91110, 2004

[16] S. Lazebnik, C. Schmid and J. Ponce, “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories”, Proc. of CVPR, 2006

[17] KTH-TIPS: *http://www.nada.kth.se/cvap/databases/kth-tips/*

[18] UIUC : *http://www-cvr.ai.uiuc.edu/ponce_grp/data/*

[19] UMD : *http://www.cfar.umd.edu/~fer/website-texture/texture.htm*